Blender na interaktivan način „Cookbook“

https://github.com/lsimic/STEMKitteens

Gdje pronaći ovaj projekt?

https://github.com/lsimic/STEMKitteens

Što ćemo koristiti?

- MicroBit
- USB kabel
- MicroBit online python editor https://python.microbit.org/v/1.1
- Blender https://www.blender.org/

Što ćemo raditi

- Upoznati se sa osnovama Blendera, alata za 3D modeliranje
- Napraviti jednostavan model u Blenderu
- Kontrolirati modele u Blenderu koristeći MicroBit

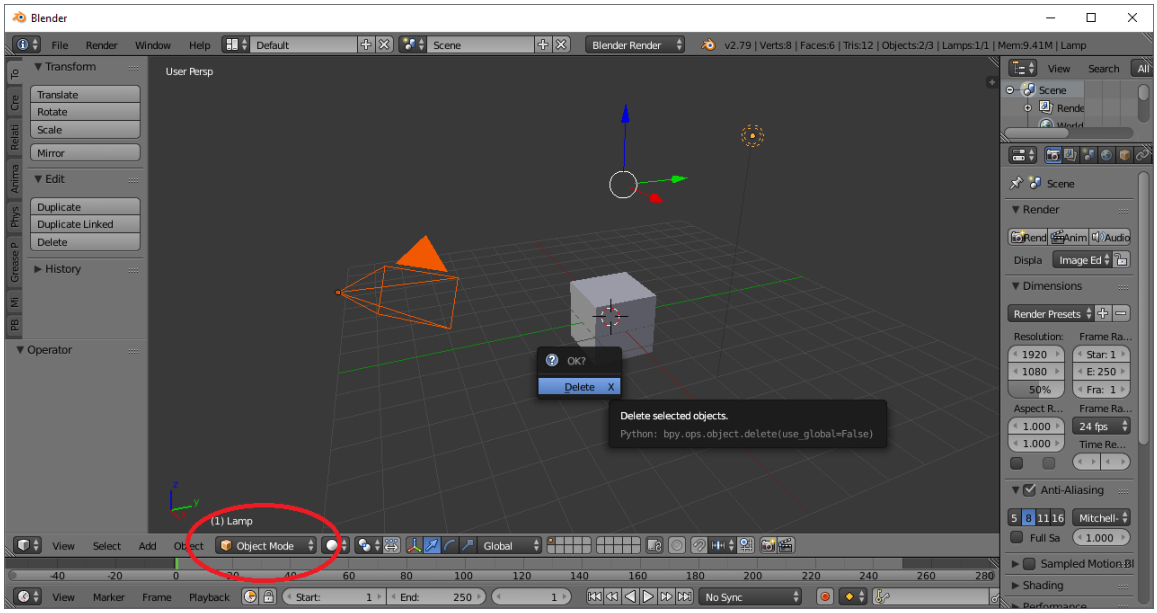
Korisni linkovi

- https://www.blender.org/support/tutorials/
- https://docs.python.org/3/tutorial/index.html
- https://microbit.org/

Blender – modeliranje

(NEOBAVEZNO) OBRISATI LAMPU I KAMERU, JER NAM NE TREBAJU

A – deselect all
C – odabrati kameru i lampu
X/del – obrisati kameru i lampu

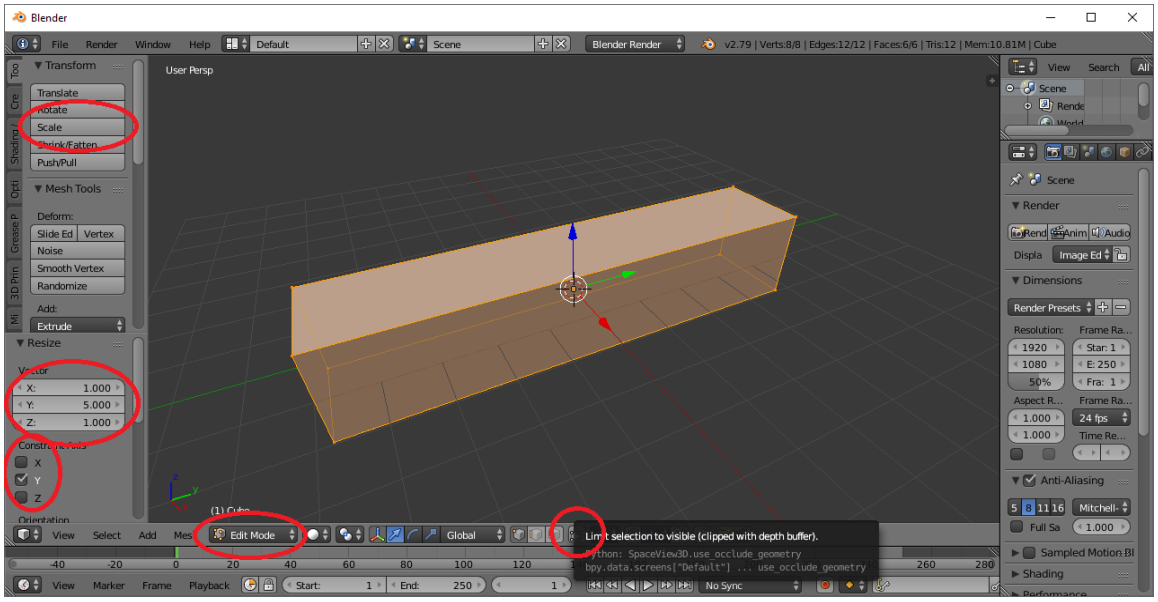


PREBACITI SE U EDIT MODE

Edit mode nam omogućava uređivanje objekata, dodavanje novih točaka itd…
Koristiti TAB, ili button zaokružen gore

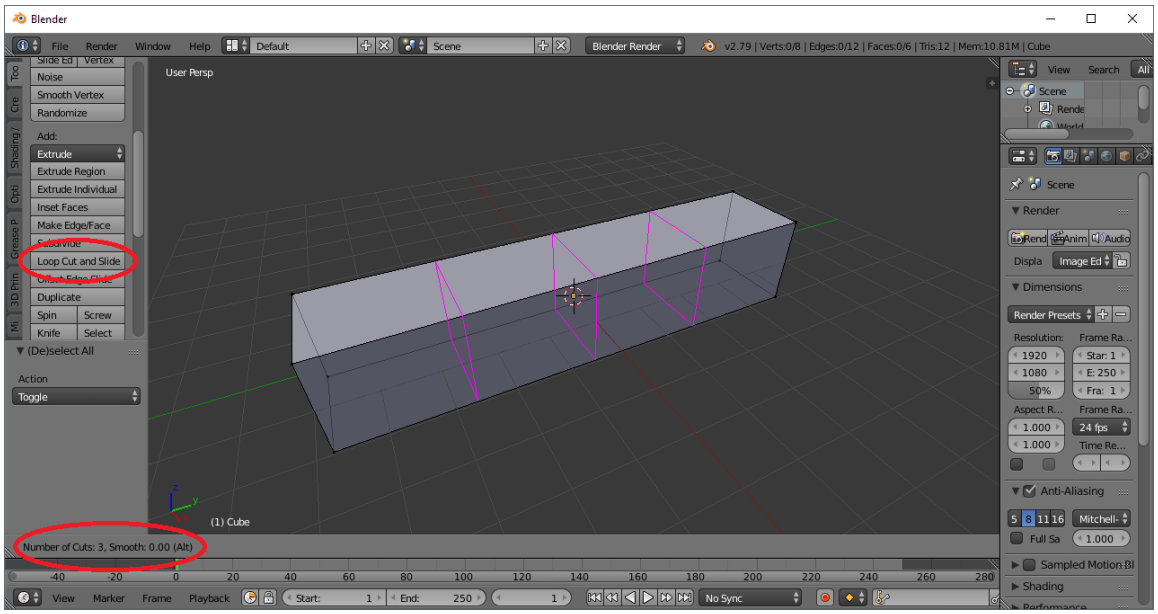
ZA LAKŠI RAD, ODABRATI LIMIT SELECTION TO VISIBLE
POVEĆATI POČETNU KOCKU 5X NA Y OSI

A – select all
S ili ui button (scale) + Y (Y os) + 5 (5 puta)
Enter (kako bi potvrdili)

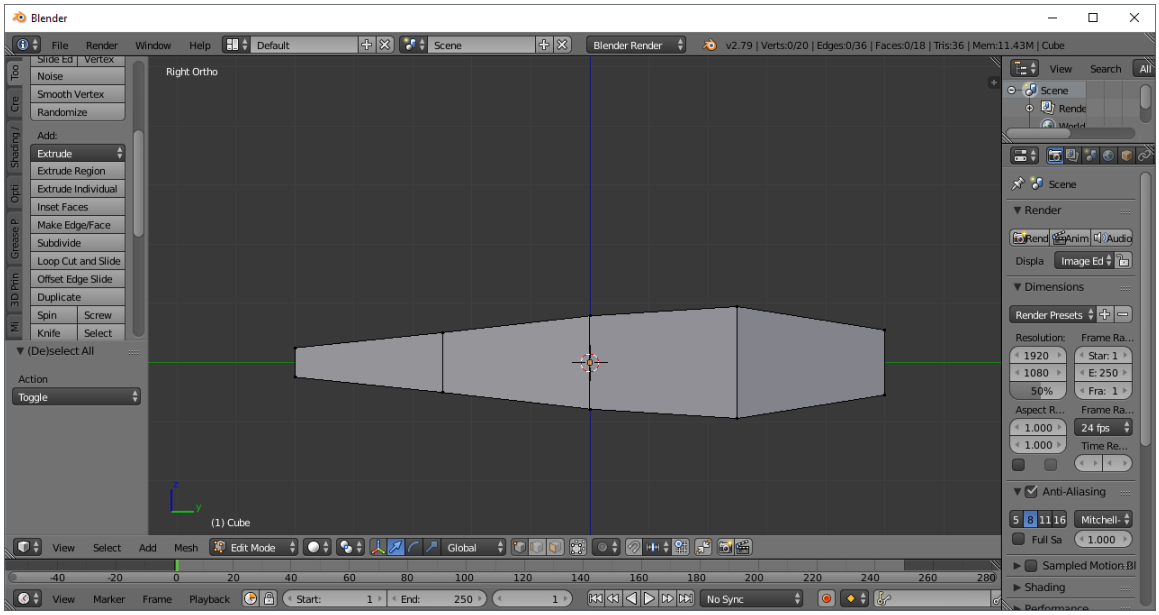


LOOP CUT AND SLIDE

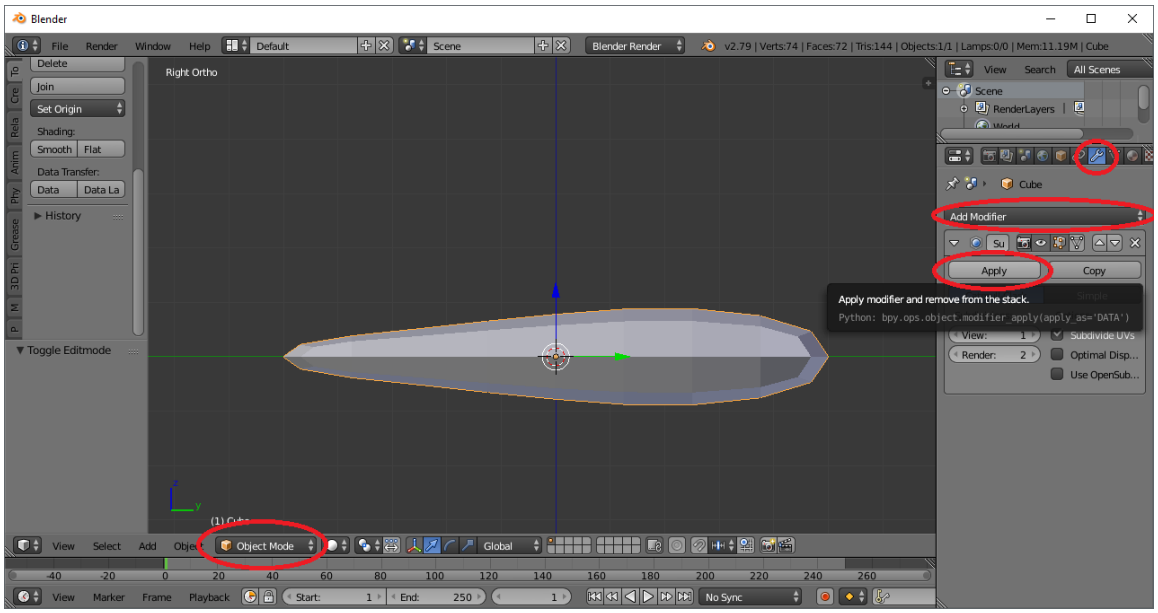
CTRL+R ili UI button – loop cut and slide
SCROLL – povećati/smanjiti broj rezova – 3 reza (3 ružičaste linije)
LMB – potvrditi



SVAKI „LOOP“ MALO SMANJITI, KAKO BI DOBILI NEŠTO SLIČNO OVOME NA SLICI

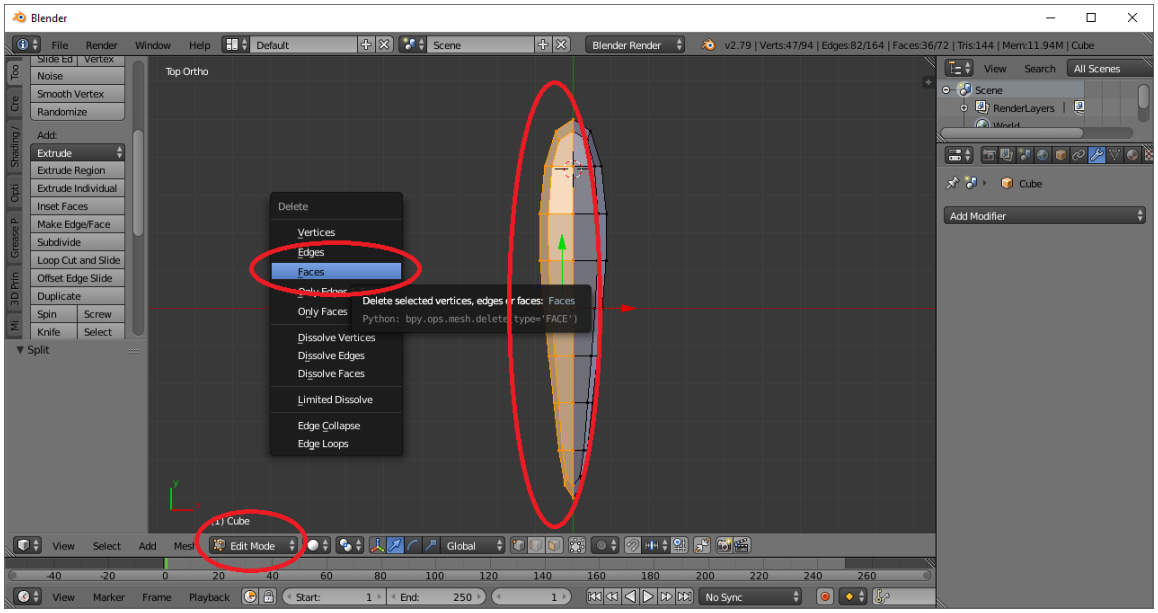


VRATITI SE U EDIT MODE, DODATI I APLICIRATI SUBDIVISION SURFACE MODIFIER

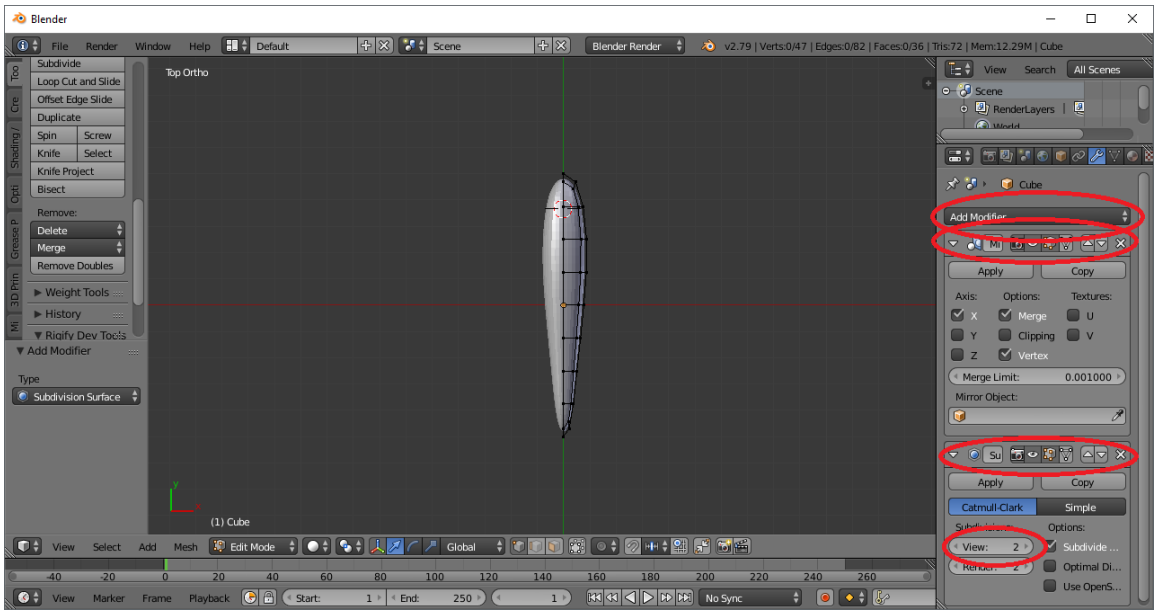


OBRISATI LIJEVU STRANU

7 – pogled od gore
C – circle select
del/X/UI button – delete faces

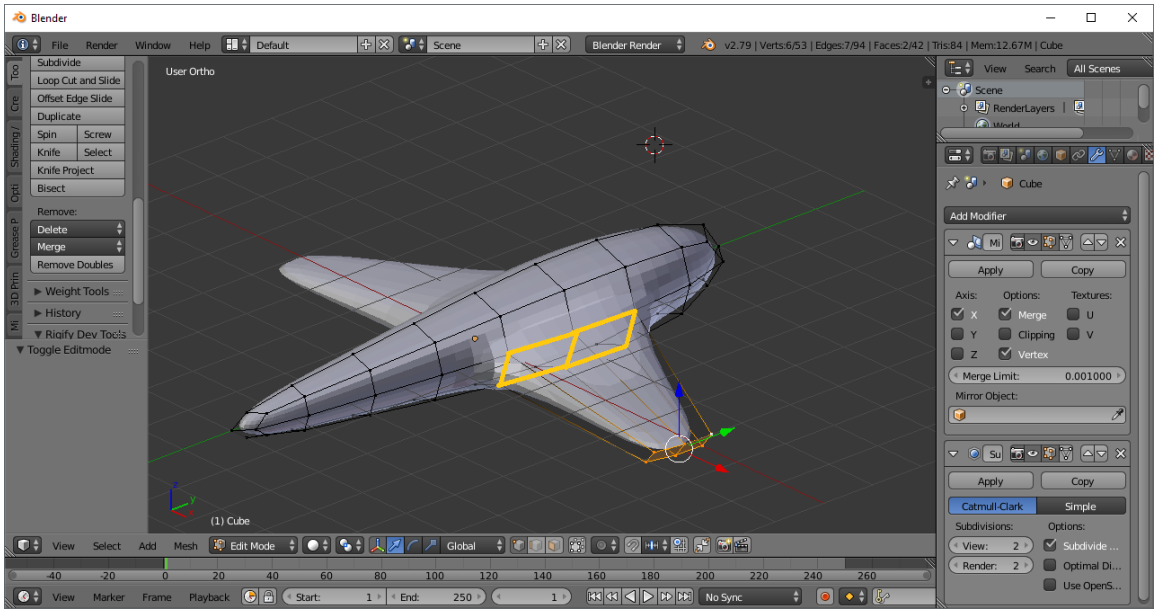


DODATI MIRROR I SUBDIVISION MODIFIER

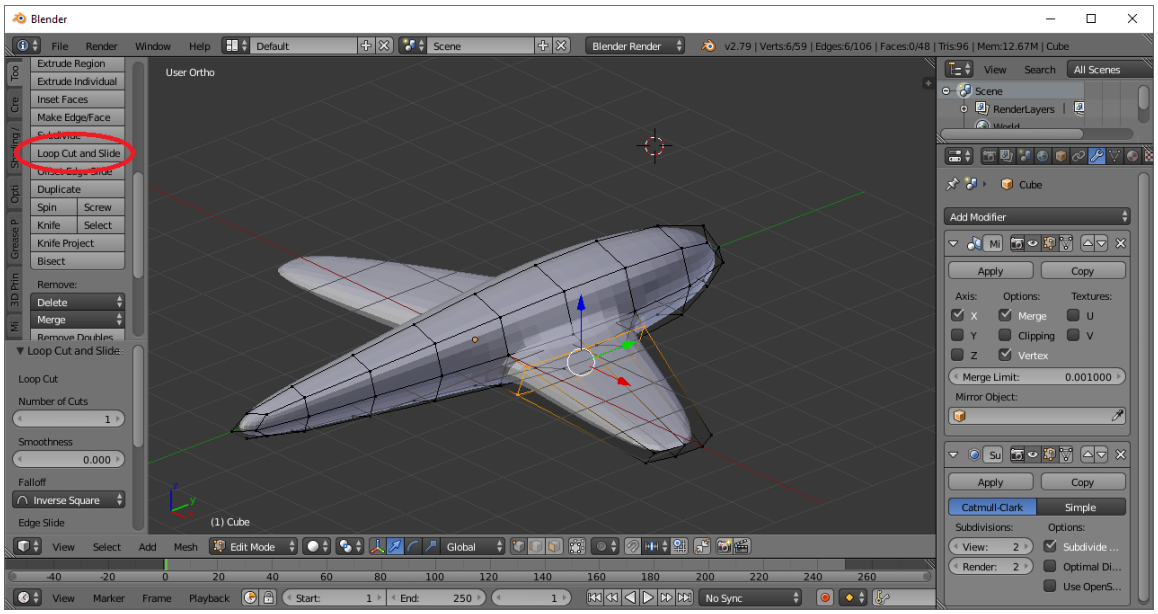


DODATI KRILA

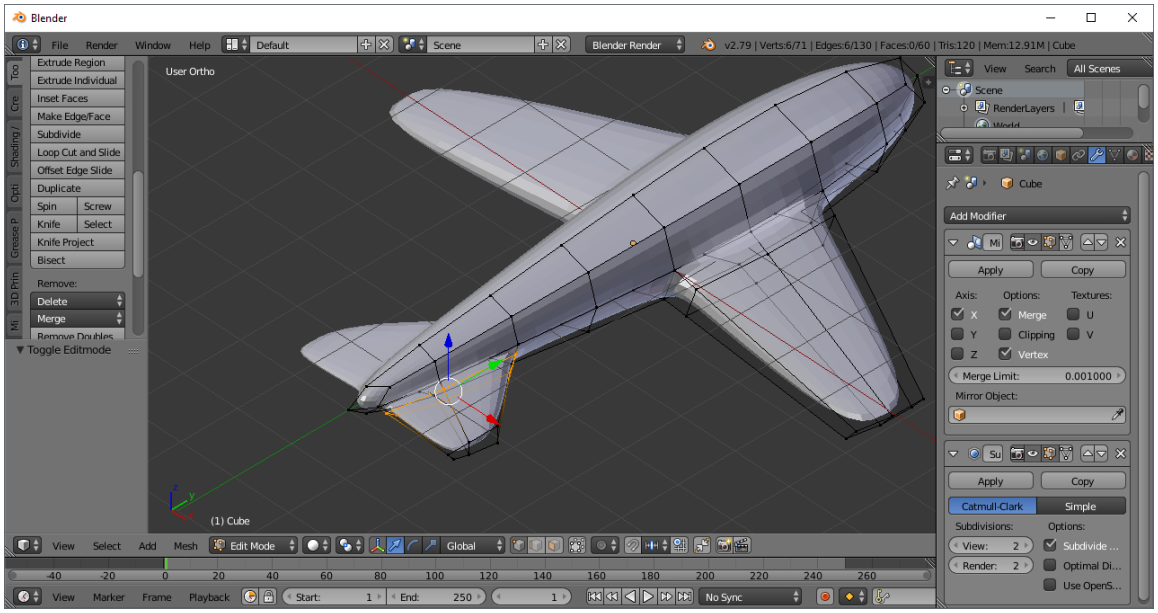
E – extrude, odabrati x os
S – scale, prvo po Z, zatim po Y osi
G – grab ili klik+drag strelice – pomaknuti po y osi.



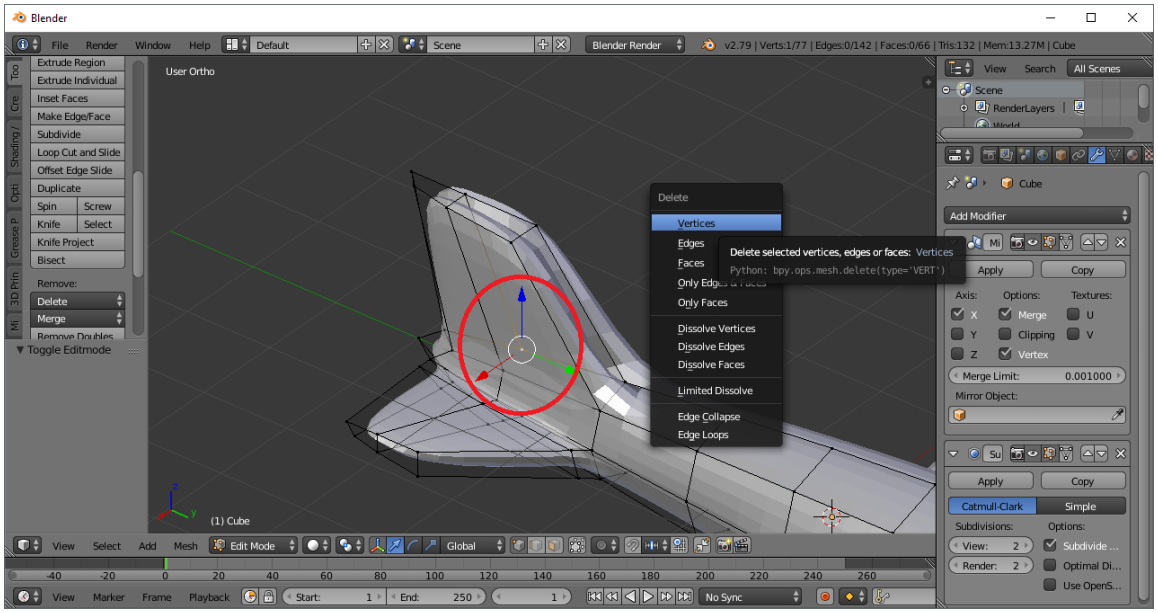
Dodati još jedan loop cut, kako bi dobili ljepši oblik



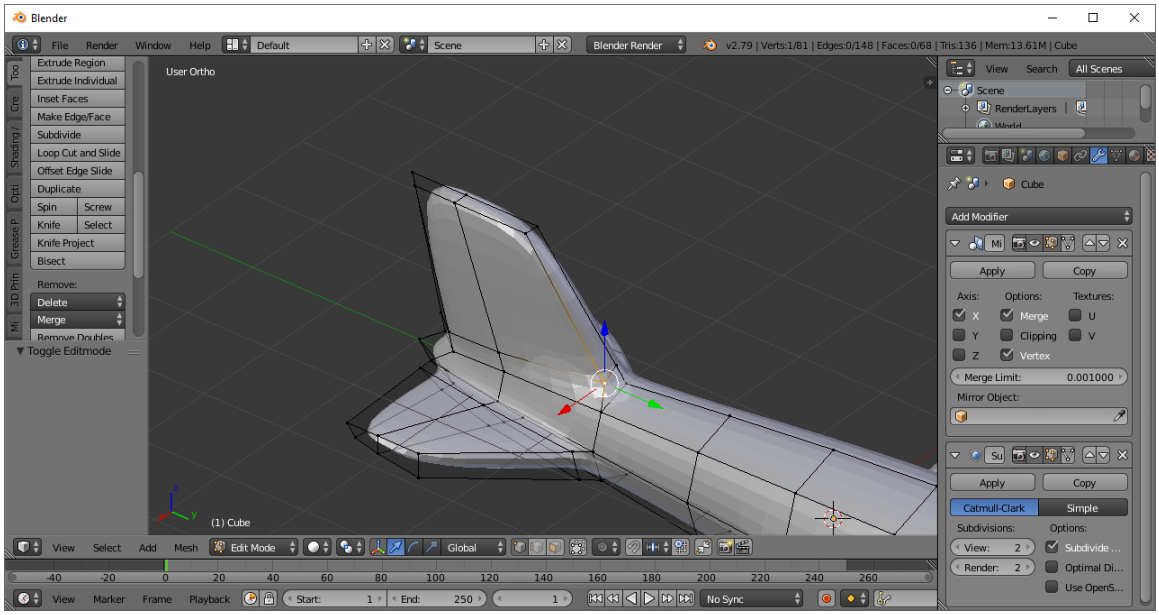
PONOVIMO ISTU STVAR ZA STRAŽNJA KRILA



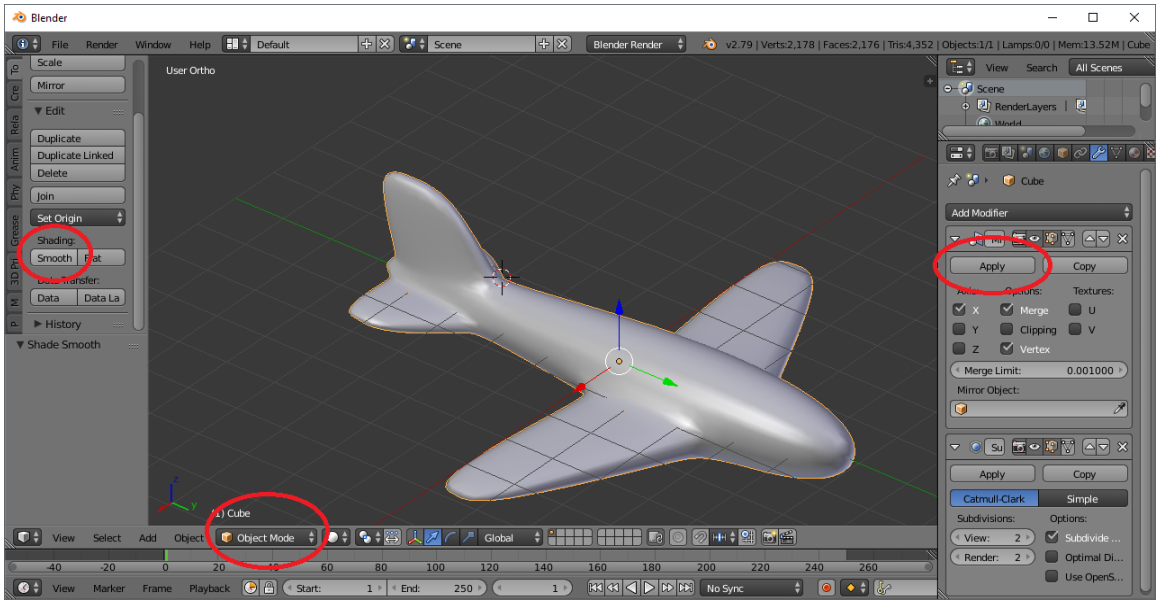
NA SLIČAN NAČIN RADIMO I REP, SAMO ŠTO MORAMO OBRISATI JEDNU TOČKU KOJA NAM SMETA…



JOŠ MOŽEMO MALO POMAKNUTI TOČKE I UREDITI OBLIK



VRATITI SE U OBJECT MODE, ODABRATI SMOOTH SHADING
APLICIRATI MIRROR MODIFIER

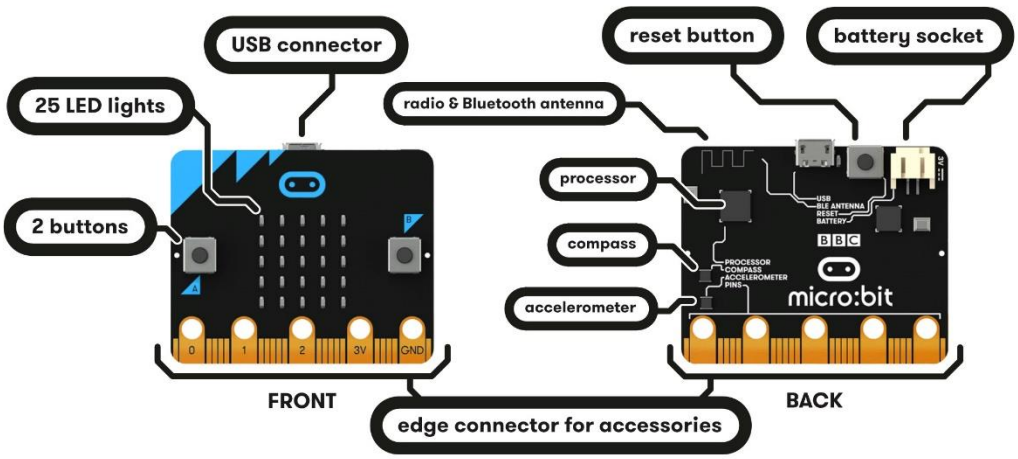


Kako kontrolirati model u Blenderu putem MicroBita – programiranje MicroBit-a

Sada ćemo vidjeti kako ćemo kontrolirati model u blenderu preko MicroBita. Kako bi to postigli prvo trebamo vidjeti koje podatke nam MicroBit može očitati sa svojih senzora, i na koji način to radi.

Zatim trebamo te podatke poslati na računalo kako bi ih mogli dalje koristiti.

OČITAVANJE PODATAKA SA SENZORA



MicroBit na sebi sadrži brojne senzore, no mi ćemo se sada usredotočiti na jedan od njih – akcelerometar[1]

Uloga akcelerometra je da mjeri ubrzanje nekog tijela u prostoru. Drugim riječima, akcelerometrom možemo otkriti kada se nešto pomakne ili nagne.

Vrijednosti akcelerometra dohvaćamo na sljedeći način

```
1.      x = accelerometer.get_x()
2.      y = accelerometer.get_y()
```

Microbit dohvaća podatke sa akcelerometra u rasponu od -2000 do 2000[2], No mi ćemo rijetko preći te vrijednosti jer je za nas bitno samo kako je microBit okrenut. U tom će slučaju vrijednosti biti između -1000 i 1000.

[1] https://microbit.org/guide/features/#accel
[2] https://microbit-micropython.readthedocs.io/en/latest/accelerometer.html

SLANJE PODATAKA NA RAČUNALO

Sada kada smo dohvatili vrijednosti sa senzora, trebamo ih poslati na računalo.

MicroBit je spojen na računalo putem USB-a. Ako možemo poslati naš program na MicroBit, to znači da možemo i poslati podatke sa MicroBita na računalo, ali kako?

Za komunikaciju ćemo koristiti UART[3] (universal asynchronous reciever-transmitter)

Prije nego možemo slati i primati poruke putem UART-a, potrebno ga je inicijalizirati

```
1. uart.init(9600)
```

Zatim, kada smo inicijalizirali UART, možemo putem njega poslati neku poruku

```
1.      msg = 'Hello \n'
2.      uart.write(msg)
```

SPOJIMO SVE ZAJEDNO

Još je samo ostalo formatirati naše podatke sa senzora u jedan string koji je pogodan za slanje UART-om

```
1.      msg = str(x) + ',' + str(y) + '\n'
```

Kompletan kod izgleda ovako:

```
1. from microbit import *
2.
3.
4. uart.init(9600)
5.
6. while True:
7.     x = accelerometer.get_x()
8.     y = accelerometer.get_y()
9.     msg = str(x) + ',' + str(y) + '\n'
10.      uart.write(msg)
11.      sleep(50)
```

[3] https://microbit-micropython.readthedocs.io/en/latest/uart.html

Kako kontrolirati model u Blenderu putem MicroBita – programiranje Blender-a

Zbog toga što standardna biblioteka pythona, a samim time i Blender ne uključuje API odnosno metode za komunikaciju putem UART-a potrebno je koristiti neku biblioteku koja će nam omogućiti tu funkcionalnost. Za komunikaciju na našem računalu ćemo koristiti PySerial.

PySerial nam omogućava korištenje serijske/UART komunikacije bez obzira na platformu(Windows/Linux/…)

INSTALACIJA PYSERIAL BIBLIOTEKE

Zbog toga što će se naš Python kod izvoditi unutar blendera, potrebno je na nešto drugačiji način od uobičajenog instalirati biblioteku.

Prvo ćemo preuzeti biblioteku sa sljedećeg linka(.tar.gz):

https://pypi.org/project/pyserial/#files

Zatim ćemo preuzetu datoteku raspakirati
Zatim ćemo folder „serial“ kopirati u …\Blender\2.79\python\lib
Kako bi nam PySerial postao dostupan unutar Blendera, potrebno ga je restartirati.

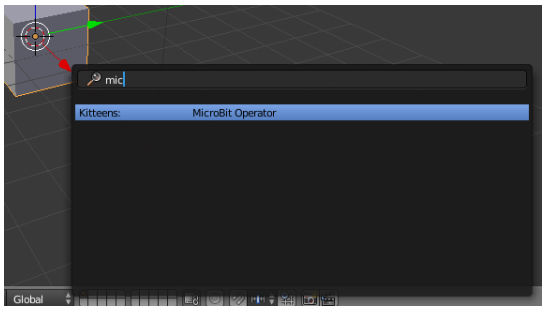
ŠTO JE BLENDER OPERATOR

Operatori u blenderu su python skripte koje izvršavaju određene zadaće. Pomoću operatora možemo na različite načine manipulirati s objektima.

Blender dolazi sa brojnim već gotovim operatorima, kojima možemo npr. Pomicati i rotirati objekte.

Uz operatore koji dolaze s blenderom, mi možemo pisati vlastite operatore kako bi proširili mogućnosti koje blender pruža.

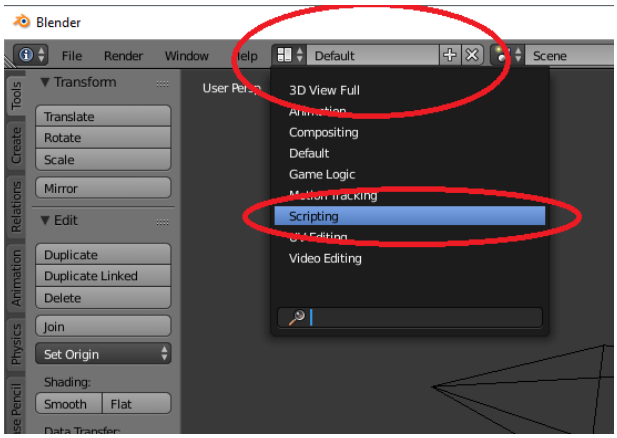
Kada registriramo operator unutar blendera, on će biti dostupan putem „spacebar menu“. Uz to možemo dodati i korisničko sučelje za naš operator



PISANJE KODA UNUTAR BLENDERA

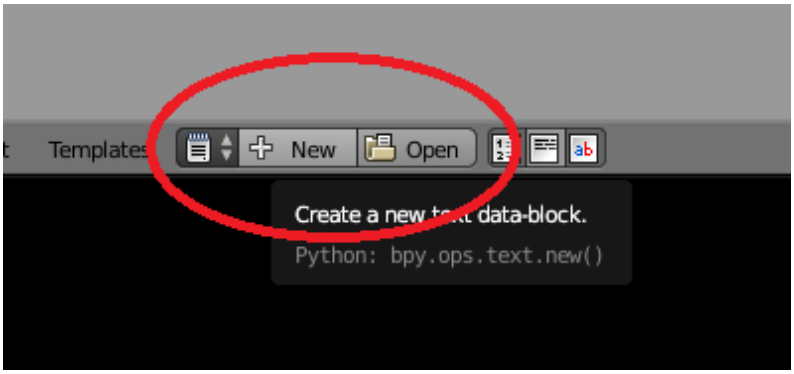
Dobra strana blendera je to što ga je vrlo jednostavno proširiti i dodati mu različite funkcionalnosti koje možda developeri nisu originalno predvidjeli, a nama baš trebaju, ili za male, zanimljive projekte poput ovoga. To možemo napraviti tako da napišemo vlastite, ili iskoristimo jedan od brojnih python addona koji postoje za blender.

Za manje projekte, blender u sebi sadrži code/text editor i python konzolu koje možemo koristiti za pisanje koda i isprobavanje raznih funkcionalnosti.



Kako bismo ubrzali postupak pisanja koda, preskočit ćemo pisanje nekih dijelova koji nisu toliko zanimljivi i ne sadrže u sebi nikakvu posebnu logiku, ali su potrebni za rad. To je kod koji odrađuje nekakvu „inicijalizaciju“ i slično…

Prvo ćemo stvoriti novu python skriptu unutar blendera, ili otvoriti postojeći template koji smo preuzeli sa githuba.



Napomena: pišemo kod unutar datoteke u kojoj se nalazi i naš model.
Model također možemo i dodati naknadno, putem file->append.[4]

[4] https://docs.blender.org/manual/en/latest/data_system/linked_libraries.html

TEMPLATE:

```
1.  import bpy
2.  import math
3.  import serial
4.
5.
6.  class MicroBitOperator(bpy.types.Operator):
7.      bl_idname = "kitteens.micro_bit_operator"
8.      bl_label = "MicroBit Operator"
9.
10.     _timer = None
11.
12.     def do_stuff(self, context):
13.         # code goes here
14.
15.
16.     def modal(self, context, event):
17.         if event.type == 'ESC':
18.             self.cancel(context)
19.             return {'CANCELLED'}
20.
21.         if event.type == 'TIMER':
22.             self.do_stuff(context)
23.
24.         return {'PASS_THROUGH'}
25.
26.     def execute(self, context):
27.         # code goes here
28.         self._timer = context.window_manager.event_timer_add(0.05, context.window)
29.         context.window_manager.modal_handler_add(self)
30.         return {'RUNNING_MODAL'}
31.
32.     def cancel(self, context):
33.         # code goes here
34.         context.window_manager.event_timer_remove(self._timer)
35.         return {'CANCELLED'}
36.
37.
38. def register():
39.     bpy.utils.register_class(MicroBitOperator)
40.
41.
42. def unregister():
43.     bpy.utils.unregister_class(MicroBitOperator)
44.
45.
46. if __name__ == "__main__":
47.     register()
```

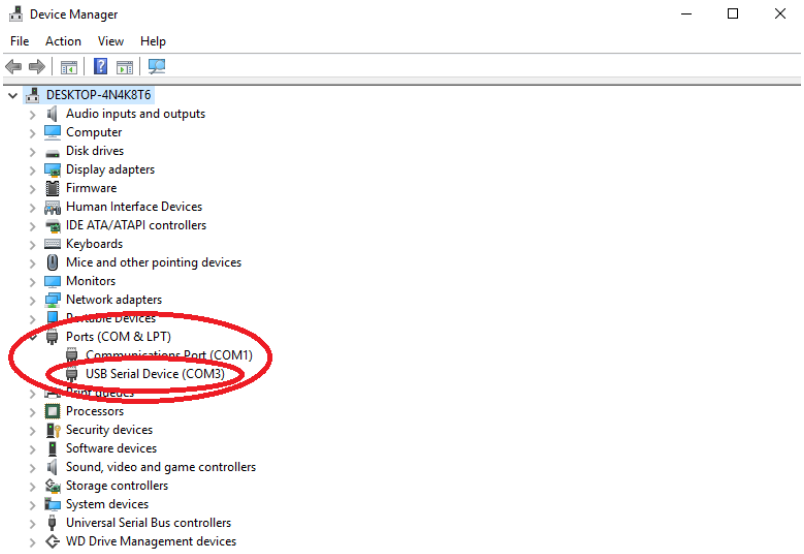
Napomena: bolje/lakše je kopirati kod s github stranice, ili preuzeti .py datoteku i otvoriti ju iz blendera.

USPOSTAVLJANJE KOMUNIKACIJE UNUTAR BLENDERA

Prvo je potrebno otvoriti serijsku komunikaciju na određenom portu.

```
1. ser = serial.Serial('COM3')
```

Port (u ovom slučaju 'COM3' možemo provjeriti u device manageru (za windows)



Nakon toga, možemo dodati kod koji će otvarati i zatvarati serijsku vezu kada budemo palili, odnosno gasili blender operator

```
1.     def execute(self, context):
2.         if not ser.is_open:
3.             ser.open()
4.         ...
```

```
1.      def cancel(self, context):
2.         if ser.is_open:
3.             ser.close()
4.          ...
```

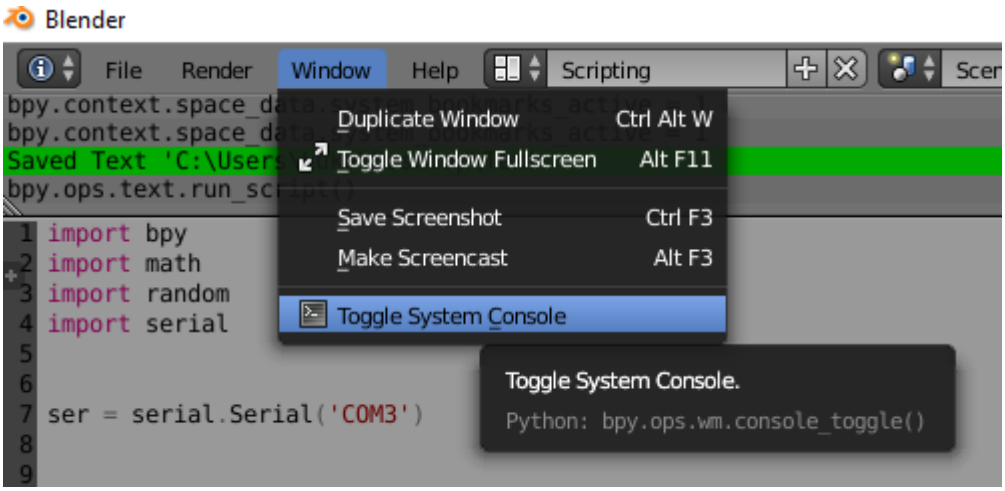
Sada u našoj do_stuff() funkciji možemo pročitati vrijednosti koje nam šalje MicroBit, i ispisati ih

```
1.         msg = ser.readline()
2.         print(msg.decode('utf-8').strip('\n'))
```

Kako bismo pročitali podatke, potrebno je poruku prvo dekodirati

vrijednosti će biti ispisane na konzoli.

Napomena: nemojte zatvarati konzolu. Kada zatvorite konzolu, s njim će se zatvoriti i blender.



DOHVAĆANJE OBJEKTA I ROTACIJA

Kada pročitamo poruku poslanu sa microbita, ona je u obliku stringa, i potrebno ju je izmjeniti i prepraviti kako bi njen sadržaj mogli iskoristiti za rotiranje našeg objekta.

Prvo trebamo rastaviti poruku na dijelove, koji su odvojeni zarezom, i ukloniti znak za novi red

```
1.          msg_parts = msg.decode('utf-8').strip('\n').split(',')
```

Zatim izračunamo vrijednosti za rotaciju.

```
1.          rot_x = math.radians(int(msg_parts[0])*90/1000)
2.          rot_y = math.radians(int(msg_parts[1])*90/1000)
3.          rot_z = 0
```

Na kraju trebamo dohvatiti naš objekt, i na njega primijeniti rotaciju koju smo izračunali

```
1.          ob = context.scene.objects["Cube"]
2.          ob.rotation_euler = (rot_x, rot_y, rot_z)
```

Sada, kada pokrenemo operator, trebali bi vidjeti da se naš objekt okreće zajedno s microbitom.